

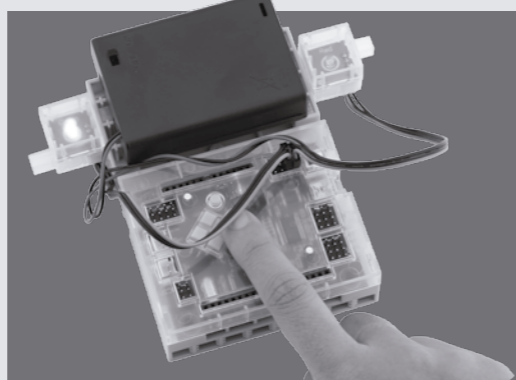
# Scratchベースで動かそう!

# Studuino スタディーノで 多電子工作

## 第2回 “あっちむいてホイ”マシン

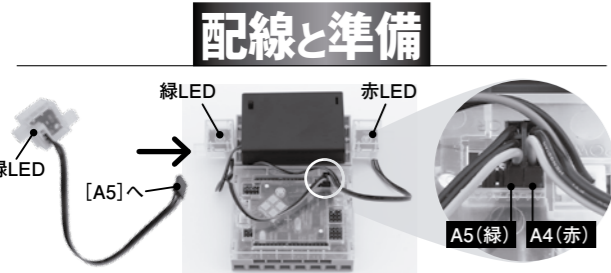
今回は、スタディーノとブロックプログラミング環境を使って、“Lチカ”デビューを果たしたね。これで、LEDを一定時間ごとに点滅させたり、ボタンを押すと点灯させたりする方法をマスターした。さらに、プログラムをスタディーノに転送して、パソコンから取り外して単独で動かすこともできるようになった。今回はこの応用として、LEDとボタンを使ったゲームをつくってみよう。つくるゲームは「あっちむいてホイ」だ!

監修・原案 / 青山学院大学客員教授 阿部和広  
協力 / NPO法人 CANVAS 文 / 塩野祐樹



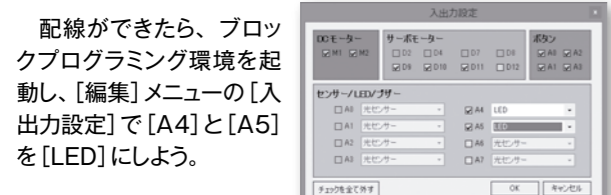
「コカねっ!」のスペシャルページで復習しよう  
[www.kodomonokagaku.com/magazine/studuino/](http://www.kodomonokagaku.com/magazine/studuino/)

「あっちむいてホイ」はみんな知っているよね。今回はその省略版として、最初のじゃんけんは省いて、顔を向ける方向は左右だけとした。スタディーノが顔を向ける方向は、2個のLEDの光で表すようにする。プレイヤーは、指を差す方向をボタンで入力することにしよう。スタディーノが顔を向けるタイミングや勝ち負けも、LEDの点滅パターンで表すぞ。



前回の配線に加えて、接続ケーブルを取り付けた緑色LEDを[A5]に取り付けよう。接続ケーブルの灰色の線が、基板の内側を向くように注意する。そして、前回の赤色LEDと同様、スタディーノの台座に写真のように差し込もう。

普通の電子部品で配線する方法もあるよ! 詳しくは71ページ下の「応用編」をチェック!



### あっちむいてホイのルール

2人でじゃんけんをして、勝った方は上下左右のどちらかを指差し、負けた方は上下左右のどちらかに顔を向ける。指差した方向と顔を向けた方向が一致したら指を差した方の勝ち、一致しなかったら、もう一度じゃんけんからやり直しとなる。

- ① じゃんけんをする
- ② じゃんけんに勝った方は上下左右のどちらかを指差し、負けた方は上下左右のどちらかに顔を向ける
- ③ 指差した方向と顔を向けた方向が一致したら、指を差した方の勝ち! 一致しなければ①からやり直し

#### 今回の特別ルール

最初のじゃんけんは省いて、向きも左右だけとする。スタディーノが左右どちらかを向くので、プレイヤーはスタディーノが向く方を当てたら勝ち、当てられなければ負け!

### 変数を使う

配線ができたら、[A4]と[A5]のLEDを1秒置きに交互に10回点滅させるプログラムを書いてみよう。前回のプログラムをちょっと変えるだけだね。

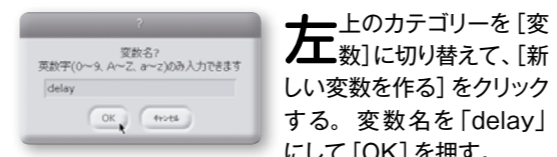
ここでは、最後の[LEDA5を消灯]に注目。これがないと、プログラムの実行が終わったときに[A5]のLEDが点いたままになるぞ。

[実行]メニューから[テストモード開始]を選びテストモードに切り替えてから、緑の旗をクリックして、ちゃんと交互に点滅することを確認しよう。

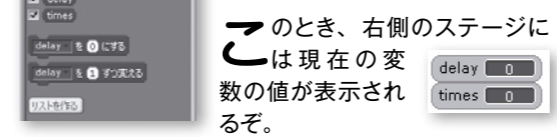
今回は、この点滅プログラムを、さまざまな点滅間隔や繰り返し回数で使うけど、そのたびに、ブロックにある「1」や「10」の数字を変えるのは面倒だ。そこで、「変数」を使ってみよう。

変数は、その言葉の通り、「変わる数」だ。それに対して、「1」や「10」のような決まった数のことを定数と呼ぶ。

変数には自由に名前をつけられるので、例えば、点滅を待つ時間なら「delay」(遅延時間)、繰り返し回数なら「times」(回数)という感じに、その中身を表す名前をつけるのとわかりやすい。Scratchでは日本語の変数名も使えたけど、ブロックプログラミング環境ではアルファベットと数字しか使えないので注意しよう。



左上のカテゴリーを[変数]に切り替えて、[新しい変数を作る]をクリックする。変数名を「delay」にして[OK]を押すと、その変数の中身を呼び出したり(参照)、変えたり(代入)するためのブロックが自動的につくられる。同じようにして「times」も追加しよう。



これらのブロックを使って、[delay]を「1」、[times]を「10」にセットし、点滅間隔と繰り返し回数をセットしたプログラムはこのようになる。緑の旗をクリックして、前のプログラムと同じ動きを確認しよう。[delay]と[times]の数字を変えると結果も変わるので、いろいろと試してみよう。

### 関数をつくる

この点滅プログラムを何回も使いたいとき、そのたびにこれだけのブロックを並べるのはかなり大変だ。また、ちょっと変更するだけでも、何か所もブロックを変えないといけなくなる。

そんなときに便利なのが「関数」だ。関数を使うと、プログラムのまとまりに名前をつけることができ、あとはその名前を指定するだけで、何回でも呼び出せるのだ。これはScratchのメッセージ送受信に似ているね\*。

\*通常のプログラミング言語の関数は、引数をとらない返り値を返す。また、数学の関数の定義(数の集合の写像)とも違うので注意しよう。

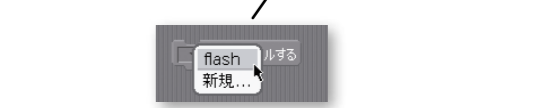
関数をつくるには、[制御]カテゴリーの[関数]ブロックを使う。「▼」をクリックするとメニューが出るので、「新規...」を選んで名前をつけよう。アルファベットと数字しか使えないのは変数と同じだ。ここでは「flash」(点滅)としよう。



この[flash関数]ブロックを、まとめたいブロックの上につける。こんな感じだ。

この関数を呼び出すには[関数をコールする]を使う。「▼」でメニューを出し、「flash」を選ぼう。これを[制御スタート]の下にある、変数の代入ブロックの下につければ完成だ。

右上の緑の旗をクリックして、前のプログラムと同じ動きを確認しよう。これで、あっちむいてホイをつくる準備ができた。



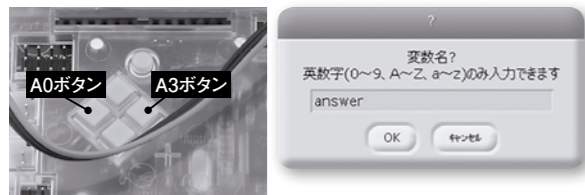
## ゲーム開始の予告

ゲームの開始と同時に、スタディーノがいきなり方向を示したら困るので、開始の予告が出るようにしましょう。これはカメラのセルフタイマーのように、最初はLEDがゆっくり点滅して、直前に速く点滅するようにすればよいだろう。それから1回LEDが消えてゲームがスタートする。プログラムはこんな感じになる。



## プレイヤーの入力

続いて、プレイヤーが指を差す方向の入力だ。これはA3ボタンが押されたら右、A0ボタンが押されたら左としよう。答えは変数「answer」(答え)に入れるので、さっきと同じ要領であらかじめつくっておこう。



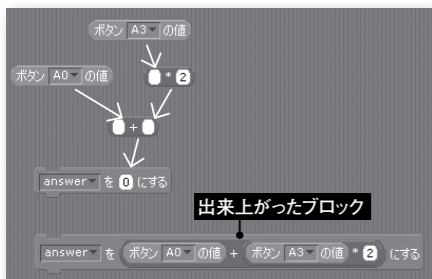
前回、ボタンでLEDを操作したとき、ボタンを押していないときは「センサー・ボード」の値が1、押しているときは0だったのを覚えているだろうか。この値と、足し算、かけ算をうまく組み合わせると、2つのボタンの押され方を1つの数字で表せる。

ボタンA0の値 + (A3の値 \* 2)

※コンピューターの世界では、「+×÷」の記号は、「+\*/」で表す。

	1 (A0ボタンが押されていない)	0 (A0ボタンが押された)
1 (A3ボタンが押されていない)	3	2
0 (A3ボタンが押された)	1	0

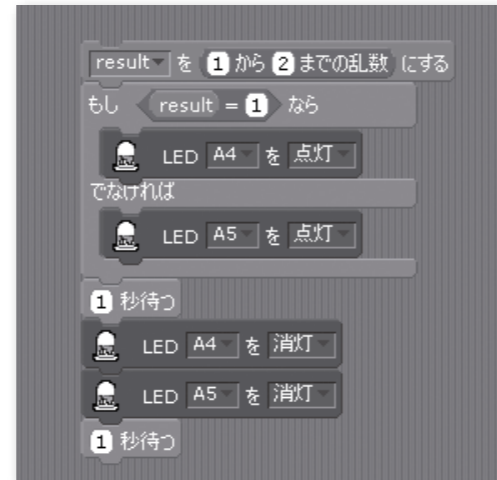
上の式で計算した結果をまとめると、表のようになる。つまり、ボタンを両方とも押すと0、右(A3)だけ押すと1、左(A0)だけ押すと2、どちらも押していないときは3となる。これをプログラムにすると左のようになる。足し算やかけ算は「演算」カテゴリにある。式のカッコ(計算する優先順位)は、ブロックを重ねる順番で表されることに気をつけよう。



## スタディーノの向き

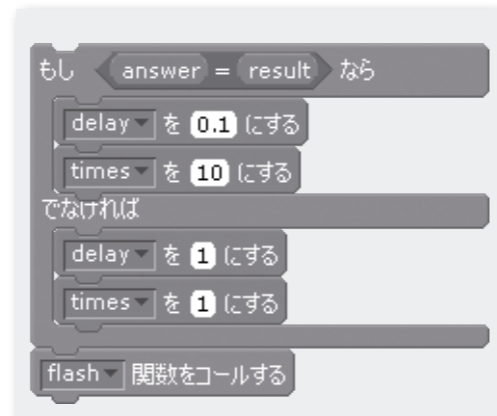
スタディーノの向きは、乱数で決定して、それに従ってLEDを点灯すればよい。乱数とは、偶然に得られるランダムな数のこと。ランダムなので、次にどんな数が出るかはわからない。ここでは、1か2のどちらかが得られるようにしている。乱数で得られた値は、変数「result」(結果)に入れるので、この変数もあらかじめつくっておこう。

プログラムはこんな感じになる。乱数で1が得られたら右の[A4]のLEDが、2なら左の[A5]のLEDが光った後、1秒後に消えるようにしている。



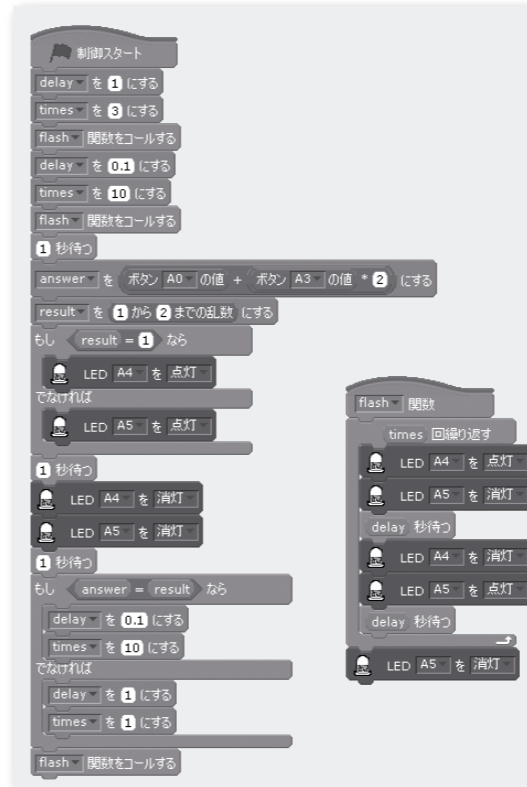
## 勝ち負けの判定

プレイヤーの入力と、スタディーノが示した向きが同じならプレイヤーの勝ち、違っていたら負けとなる。ボタンを押さなかったときや、両方のボタンを押したときも、プレイヤーの負けだ。勝ちの場合は速い点滅、負けの場合は遅い点滅で表そう。



## プログラムの完成と転送

以下にプログラムの全体図を示す。



プログラムが完成したら、スタディーノに転送しよう。前回やった通り、[実行]から[プログラム作成・転送]を選び、ダイアログが消えるのを待とう。転送が終わったら、ゲームが自動でスタートする。もう一度やりたいときは、リセットボタンを押そう。

このとき、テストモードとプログラム転送では、点滅の仕方が違うことに気づいただろうか。テストモードでは、点滅が指定した数値よりもゆっくりで、その間隔もちょっとばらばらだ。つまり、正確な動きを知るためには、テストモードだけでなく、プログラム転送も行ってみよう。

## 電池ボックスを取り付けて

電力カバー台座に寄せ、USBケーブルを外せば、携帯ゲームとして使えるぞ(電池ボックスが落ちやすいので輪ゴムなどで止めてもよい)。家族や友達とやってみよう。また、どうすれば、もっとおもしろくなるか考えて、プログラムを改造してみよう。

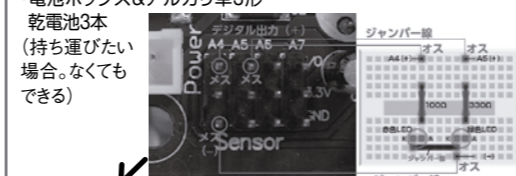


## 応用編 スタディーノ基板と普通の電子部品でやってみよう



用意するもの  
 ・スタディーノ基板  
 ・LED (赤1個、緑1個)  
 ・抵抗 (100Ω1個、330Ω1個)  
 ・ジャンパー線 (メス-オス 3本、オス-オス 1本)  
 ・mini USBケーブル  
 ・電池ボックス&アルカリ単3形乾電池3本 (持ち運びたい場合。なくても可)

より安い部品でつくりたい&電子工作好きな人向けの応用編だ。連載ではアーテック社のLEDを使っているけれど、スタディーノ基板と電子工作で使う普通のLEDや部品を使って、同じことをやってみよう。単体のスタディーノ基板は、各種オンラインショップで2000円〜3000円前後で売られているよ。あとはブレッドボード、抵抗、LEDを電子部品ショップで購入してね。



◀配線できたら、あとは今回紹介した内容でプログラムを組み立ててね。同じように遊べるよ。  
 ▲基板の左下「A4」「A5」にそれぞれ3つ端子がある。一番上をプラス、一番下をマイナスと考えて、上のように配線しよう。

## KoKaスタディーノ基本セットがヨドバシドットコムで販売中

前回の「Lチカ」と、今回の「あっちむいてホイ」が実際にできるキットがヨドバシドットコムで販売中だ。初めての人にはぴったりだから、ぜひやってみてね。



4100円(税込) ◀1/20までの割引価格!!  
 ヨドバシドットコム [www.yodobashi.com/](http://www.yodobashi.com/)  
 ※「KoKaスタディーノ基本セット」で検索!